\indexentry{alpha|(bold}{1}
\indexentry{alpha|)}{5}

will become

\item alpha, \bold{1--5}

A couple of special cases are worth mentioning here. First, entries like

\indexentry{alpha|(}{1}
\indexentry{alpha|bold}{3}
\indexentry{alpha|)}{5}

will be interpreted as

\item alpha, \bold{3}, 1--5

but with a warning message in the transcript about the encounter of an inconsistent page encapsulator. Secondly, an explicit range beginning in a Roman page number and ending in Ababic is considered an error. In a case like this the range is broken into two subranges, if possible, one in Roman, the other in Arabic. For instnace,

\indexentry{alpha|(}{i}
\indexentry{alpha}{iv}
\indexentry{alpha}{3}
\indexentry{alpha|)}{7}

will be turned into

\item alpha, 1--iv, 3--7

with a warning message in the transcript complaining about the illegal range formation.

Finally, every special symbol mentioned in this section may be escaped by the **quote** operator ('"'). Thus

\indexentry{alpha"@beta}{1}

will actually become

\item alpha@beta, 1

as a result of executing *makeindex*. However, if **quote** is preceded by **escape** ('\'), its following letter is not escaped. That is,

\indexentry{f\"ur}{1}

means

\item f\"ur, 1

which represents umlaut accented 'u' to the TEX family of processors.

SEE ALSO

tex(l), latex(l),*qsort(3)*

AUTHOR

Pehong Chen
Computer Science Division
University of California, Berkeley
phc@berkeley.edu

THANKS

Leslie Lamport contributed significantly to the design. Michael Harrison provided valuable comments and suggestions.

considered a symbol-leading pattern instead of a number.

SPECIAL EFFECTS

In the normal case entries such as

```
\indexentry{alpha}{1}
\indexentry{alpha!beta}{3}
\indexentry{alpha!beta!gamma}{10}
```

in the raw index file will be converted to

```
\item alpha, 1
  \subitem beta, 3
    \subsubitem gamma, 10
```

in the output index file by *makeindex*. Notice that the **level** symbol ('!') is used to delimit levels of nesting.

It is possible to make an item appear in a designated form by using the **actual** ('@') operator. For instance,

```
\indexentry{alpha@{\it alpha\/}}{1}
```

will become

```
\item {\it alpha\/}  1
```

after the conversion. The idea is that the pattern preceding '@' is used as sort key, whereas the one following it is put in the actual result. However, the same key with and without the **actual** part are regarded as distinct entries.

It is also possible to encapsulate a page number with a designated command using the **encap** ('|') operator. For example, in the default case,

```
\indexentry{alpha|bold}{1}
```

will be converted to

```
\item alpha \bold{1}
```

where \bold{n} will expand to {\bf n}. This allows the **encap** operator to be used to set pages in different fonts, thereby conveying more information about whatever being indexed. For instance, given the same key the page where its definition appears can be in one font while where its primary example is given can be in another, with other ordinary appearances in a third. Notice that in this example, the three output attributes associated with page encapsulation **encap_prefix**, **encap_infix**, and **encap_suffix** correspond respectively to backslash, left brace, and right brace. If this is to be formatted by languages other than LaTeX, they would be defined differently.

By the same token, the **encap** operator can be used to make cross references in the index. For instance,

```
\indexentry{alpha|see{beta}}{1}
```

will become

```
\item alpha \see{beta}{1}
```

in the output index file after the conversion, where

```
\see{beta}{1}
```

will expand to

```
{\it see\/} beta
```

Notice that in a cross reference like this the page number disappears. Therefore, where to insert such a command in the source is immaterial.

A pair of **encap** concatenated with **range_open** ('|(') and with **range_close** ('|)') creates an explicit page range. That is,

```
\indexentry{alpha|(}{1}
\indexentry{alpha|)}{5}
```

will become

```
\item alpha, 1--5
```

Intermediate pages indexed by the same key will be merged into the range implicitly. This is especially useful when an entire section about a particular subject is to be indexed, in which case only the range opening and closing operators need to be inserted at the beginning and end of the section, respectively.

This explicit page range formation can also include an extra command to set the page range in a designated font. Thus

encap_infix <string> "{"
The prefix for the command which encapsulates the page number.

encap_suffix <string> "}".
The prefix for the command which encapsulates the page number.

line_max <number> 72
The maximum length of a line in the output beyond which a line wraps around.

indent_space <string> "\t\t"
The space to be inserted in front of a wrapped line. Default is two tabs.

indent_length <number> 16
The length of **indent_space**. In the default case this is 16 (for 2 tabs).

EXAMPLE

The following example shows a style file called *book.isty* which defines a stand-alone index for a book. By stand-alone, we mean it can be formatted independent of the main source.

```
preamble
"\\documentstyle[12pt]{book}
\\begin{document}
\\begin{theindex}
{\\small\n"

postamble
"\n\n}
\\end{theindex}
\\end{document}\n"
```

Suppose a particular book style requires the index (as well as any chapters) to start from an odd page number. Given *foo.idx* as the raw index file, the following command line produces an index in file *foo-.ind*.

makeindex –s book.isty –o foo-.ind –p odd foo

The reason to use a non-default output file name is to avoid clobbering the source output (presumably *foo.dvi*) because if the index is in file *foo.ind*, its output will also be in *foo.dvi* as a result of separate formatting using LaTeX. In the example the index is in *foo-.ind*, its output will be in *foo-.dvi* and thus introduces no confusion.

ORDERING

By default *makeindex* assumes *word ordering*. The **–l** option turns it into *letter ordering*. The only difference is whether a blank is treated as an effective letter or not. In word ordering, a blank precedes any letter in the alphabet, whereas in letter ordering, it doesn't count at all. This is best illustrated by the following example:

word order	*letter order*
sea lion	seal
seal	sea lion

Numbers are sorted in numeric order. For instance,

9 (nine), 123
10 (ten), see Derek, Bo

Letters are first sorted with uppercase and lowercase considered identical; then, within identical words the uppercase letter precedes its lowercase counterpart.

Patterns lead by a special symbol precede numbers, which precede patterns lead by a letter. The symbol here refers to anything not in the union of digits and English alphabet. This includes those which follow ´z´ in the ASCII chart. As a special case, anything started with a digit but mixed with non-digits is

lethead_prefix <string>
""
The header prefix to be inserted before a new letter begins.

lethead_suffix <string>
""
The header suffix to be inserted before a new letter begins.

lethead_flag <string>
0
The flag indicating the condition of inserting new letter header. Default is 0, which means no header. Positive means insert an uppercase letter between prefix and suffix. Negative means insert an lowercase letter.

item_0 <string>
"\n \\item "
The command to be inserted between two primary (level 0) items.

item_1 <string>
"\n   \\subitem "
The command to be inserted between two secondary (level 1) items.

item_2 <string>
"\n   \\subsubitem "
The command to be inserted between two level 2 items.

item_01 <string>
"\n   \\subitem "
The command to be inserted between a level 0 item and a level 1 item.

item_x1 <string>
"\n   \\subitem "
The command to be inserted between a level 0 item and a level 1 item. The difference between this and previous is that in this case the level 0 item doesn’t have any page numbers.

item_12 <string>
"\n   \\subsubitem "
The command to be inserted between a level 1 item and a level 2 item.

item_x2 <string>
"\n   \\subsubitem "
The command to be inserted between a level 1 item and a level 2 item. The difference between this and previous is that in this case the level 1 item doesn’t have any page number.

delim_0 <string>
", "
The delimiter to be inserted between a level 0 key and its first page number. Default is a comma followed by a blank.

delim_1 <string>
", "
The delimiter to be inserted between a level 1 key and its first page number. Default is a comma followed by a blank.

delim_2 <string>
", "
The delimiter to be inserted between a level 2 key and its first page number. Default is a comma followed by a blank.

delim_n <string>
", "
The delimiter to be inserted between two page numbers for the same key in any level. Default is a comma followed by a blank.

delim_r <string>
"--"
The delimiter to be inserted between the starting and ending page numbers of a range.

encap_prefix <string>
"\\"
The prefix for the command which encapsulates the page number.

‘%’ is a comment. The following is a list of all the specifiers and their respective arguments where <string> is an arbitrary string delimited by double quotes ("..."), <char> is a single letter embraced by single quotes (’...’), and <number> is a nonnegative integer. The maximum length of a <string> is 144. Notice that a backslash must be escaped (by an extra backslash) in the string quotation. Anything not specified in the style file will be assigned a default value, which is shown at the rightmost column. This file can reside anywhere in the path defined by the environment variable INDEXSTYLE.

% Input Style Specifiers

keyword <string> — "\\indexentry"
This is the command which tells *makeindex* that its argument is an index entry.

arg_open <char> — ´{´
This is the opening delimiter for the index entry argument.

arg_close <char> — ´}´
This is the closing delimiter for the index entry argument.

range_open <char> — ´(´
The opening delimiter indicating the beginning of an explicit page range.

range_close <char> — ´)´
The closing delimiter indicating the end of an explicit page range.

level <char> — ´!´
The delimiter which denotes a new level of subitem.

actual <char> — ´@´
The symbol which indicates that the next entry is to appear in the actual index file.

encap <char> — ´|´
The symbol which indicates that the rest of the argument list is to be used as the encapsulating command for the page number.

quote <char> — ´"´

escape <char> — ´\\´
The symbol which escapes the next letter, unless its preceding letter is **escape**. In other words, **quote** is used to escape the letter whcih immediately follows it. But if it is preceded by **escape**, it does not escape anything. Notice that the two symbols must be distinct.

% Output Style Specifiers

preamble <string> — "\\begin{theindex}\n"
The preamble of actual index file.

postamble <string> — "\n\n\\end{theindex}\n"
The postamble of actual index file.

setpage_prefix <string> — "\n \\setcounter{page}{"
The prefix of the command which sets the starting page number.

setpage_suffix <string> — "}\n"
The suffix of the command which sets the starting page number.

group_skip <string> — "\n\n \\indexsapce\n"
The vertical space to be inserted before a new group begins.

NAME

makeindex – a general purpose, formatter-independent index processor

SYNOPSIS

makeindex [–**ilqrc**] [**-s** *sty*] [**-o** *ind*] [**-t** *log*] [**-p** *no*] [*idx0 idx1 idx2 ...*]

DESCRIPTION

makeindex is a general purpose index processor. It takes one or more raw index files (normally generated by a formatter), sorts the entries, and produces the actual index file. It is not dependent on any particular format of raw index file, although the *.idx* file generated by LaTeX is default. Up to three levels (0, 1, and 2) of subitem nesting within the same entry is supported. The input format may be redefined in a style file so that raw index or glossary output from other formatters may be processed. The style file also defines the style of output index file. Unless specified otherwise, the file name base of the first input file (*idx0*) is used to determine other related input/output files. For each input file name specified, the name itself is first used. If not found and the name has no extension part, it is concatenated with the *.idx* extension. If this again fails, the program aborts.

OPTIONS

–i Use *stdin* as the input file. When this option is specified and the **–o** is not, output is written to *stdout*.

–l Use letter ordering. Default is word ordering (explained in the **ORDERING** section).

–q Quiet mode, send no messages to *stderr*. By default progress and error messages are sent to *stderr* as well as the transcript file. The **–q** option disables the *stderr* messages.

–r Disable implicit page range formation. By default three or more successive pages will be automatically abbreviated as a range (e.g. 1--5). The **–r** option disables it, making the explicit range operators the only way to create page ranges (see the **SPECIAL EFFECTS** section below).

–c Enable blank compression. By default every blank counts in the index key. The **–c** option ignores leading and trailing blanks and tabs and compresses intermediate ones to a single space.

–s *sty* Take *sty* as the style file. There is no default for the style file name. The environment variable INDEXSTYLE defines the path where the style file should be found.

–o *ind* Take *ind* as the output index file. By default the file name base of the first input file *idx0* concatenated with the extension *.ind* is used as the output file name.

–t *log* Take *log* as the transcript file. By default the file name base of the first input file *idx0* concatenated with the extension *.ilg* is used as the transcript file name.

–p *no* Set the starting page number of the output index file to be *no*. This is useful when the index file is to be formatted separately. Other than pure numbers, three special cases are allowed for *no*: *any*, *odd*, and *even*. In these special cases, the starting page number is determined by retrieving the last page number from the source log file. The source log file name is determined by concatenating the file name base of the first raw index file (*idx0*) with the extension *.log*. The last source page is obtained by searching backward in the log file for the first instance of a number included in **[**...**]**. If a page number is missing or the log file is not found, no attempt will be made to set the starting page number. The meaning of each of these cases follows.

any The starting page is the last source page number plus 1.

odd The starting page is the first odd page following the last source page number.

even The starting page is the first even page following the last source page number.

STYLE FILE

The style file format is very simple. It is a list of <*specifier*, *attribute*> pairs. There are two types of specifiers (input and output). The pairs don't have to obey any particular order in the file. A line lead by